DESIGN TECHNOLOGY CORE I
DESIGNTK 530 | FALL 2024

DUKE UNIVERSITY
VIVEK RAO, PhD

Assignment I-11

Getting started with AI

This **individual** assignment prepares you to get started to work with AI and know more about how to use VSCode in a local (not on the cloud environment). You will:

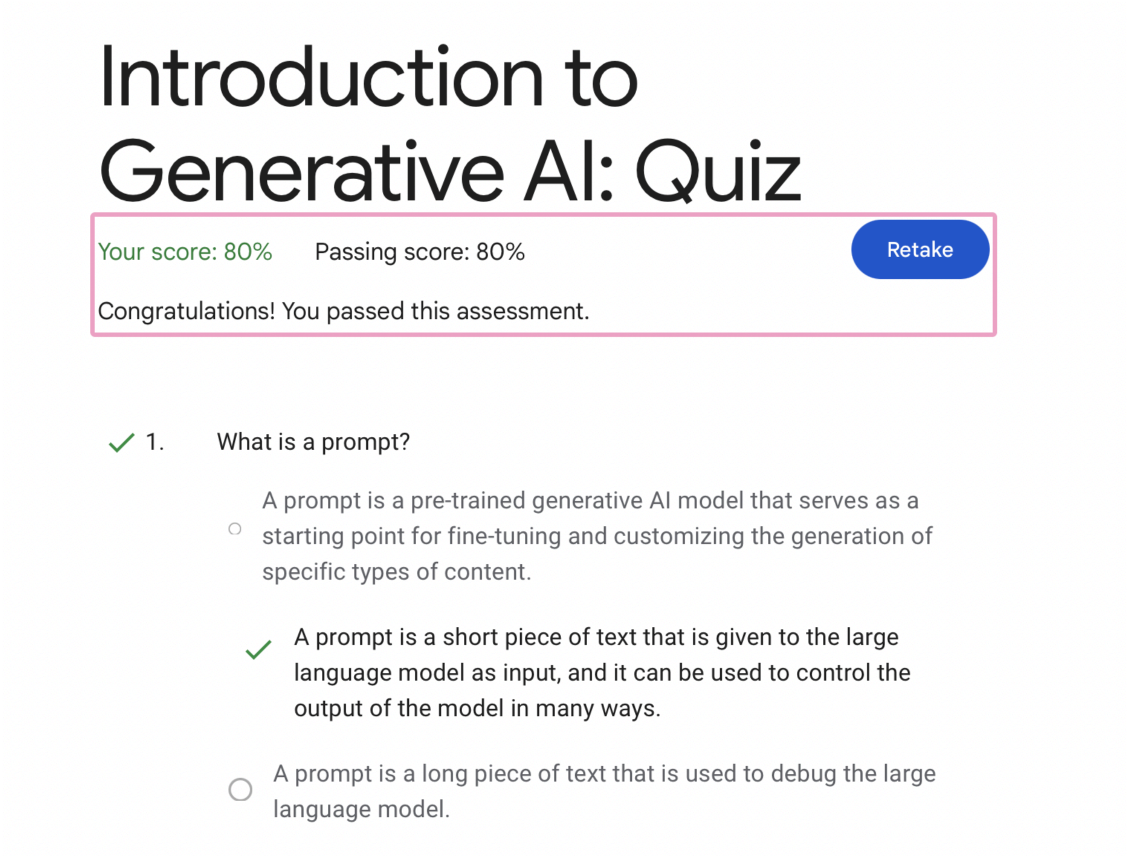
- Review a basic overview of AI and machine learning concepts, and apply these concepts to assess a product or service of your choice.
- Install appropriate packages in your local development environment.
- Recreate some basic data science activities in the local IDE.

Submit screenshots and output from VSCode as described below.

1. Follow an overview of basic AI and ML concepts.

In this section, you will review a video to learn (or review) basic concepts in AI and Machine Learning.

Problem 1.1: Watch the google cloud video here (Est: 45 min) and Attach the screenshot of your complete quiz. Please note, the last ~8 minutes of this video are a bit of a showcase / advertisement for Google's AI offerings, specifically Gemini and AI Studio.



Introduction to Generative AI: Quiz

Your score: 80%   Passing score: 80%

Congratulations! You passed this assessment.

✓ 1. What is a prompt?

○ A prompt is a pre-trained generative AI model that serves as a starting point for fine-tuning and customizing the generation of specific types of content.

✓ A prompt is a short piece of text that is given to the large language model as input, and it can be used to control the output of the model in many ways.

○ A prompt is a long piece of text that is used to debug the large language model.

Problem 1.2: Reflecting on the various types of ML algorithms covered in the video, provide two examples from your personal or professional experience where you could apply these techniques. For each example, include:

1. **Problem Statement**: Describe the specific problem you're aiming to solve.
2. **Method**: Mention the ML method you would apply and explain why this particular approach is suited to the problem.

Example 1: Pinterior app

Problem Statement**:** The goal is to create software that can analyze a user's preferred aesthetic based on images from their Pinterest boards and apply similar styles or modifications to an existing image. This involves identifying style elements in the Pinterest images and generating variations through Stable Diffusion that align with the user's aesthetic preferences.

Method:

1. Unsupervised Learning**:** To analyze and cluster the aesthetic styles from Pinterest boards, unsupervised learning techniques could be applied to extract common visual features from the board's images. These features might include color palettes, textures, or subject focus, which can be clustered to identify dominant styles or themes that characterize the user's preferences.
2. Generative AI with Stable Diffusion**:** Once the aesthetic characteristics are identified, Stable Diffusion can generate new images or modify existing ones to match the clustered style profile. By inputting the features extracted from the Pinterest images, Stable Diffusion can be guided to produce variations that are consistent with the user's preferred aesthetic.

Example 2: Stroke Busters

Problem Statement: With the stroke project that I am currently working on in Design Health I, one of the possible directions that my team might go into is to identify individuals with a high risk of stroke by analyzing risk factors including age, medical history, cholesterol levels, blood pressure, and lifestyle factors. By tracking these data with known stroke outcomes, the model could identify the early signs of potential strokes , allowing for personalized preventative measures.

Method**:** Supervised Learning with Predictive Modeling

This project will benefit greatly from predictive modeling. These methods can handle structured, tabular data (e.g., patient medical records) and can learn from labeled examples of stroke and non-stroke cases to predict the probability score of future stroke occurrences.

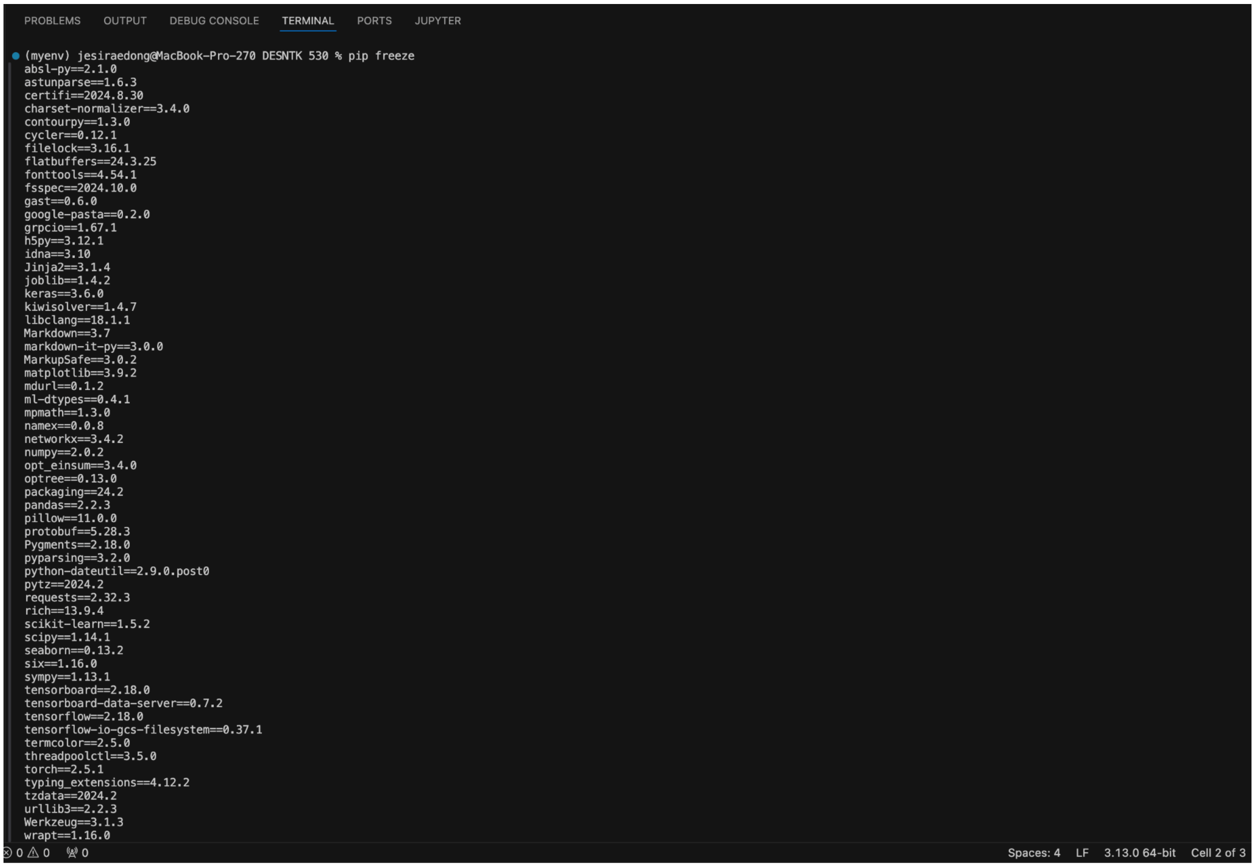
These examples should illustrate how different types of ML algorithms can be effectively applied to real-world challenges.

2. Install the necessary packages to your env to work with ML algorithms in the next class

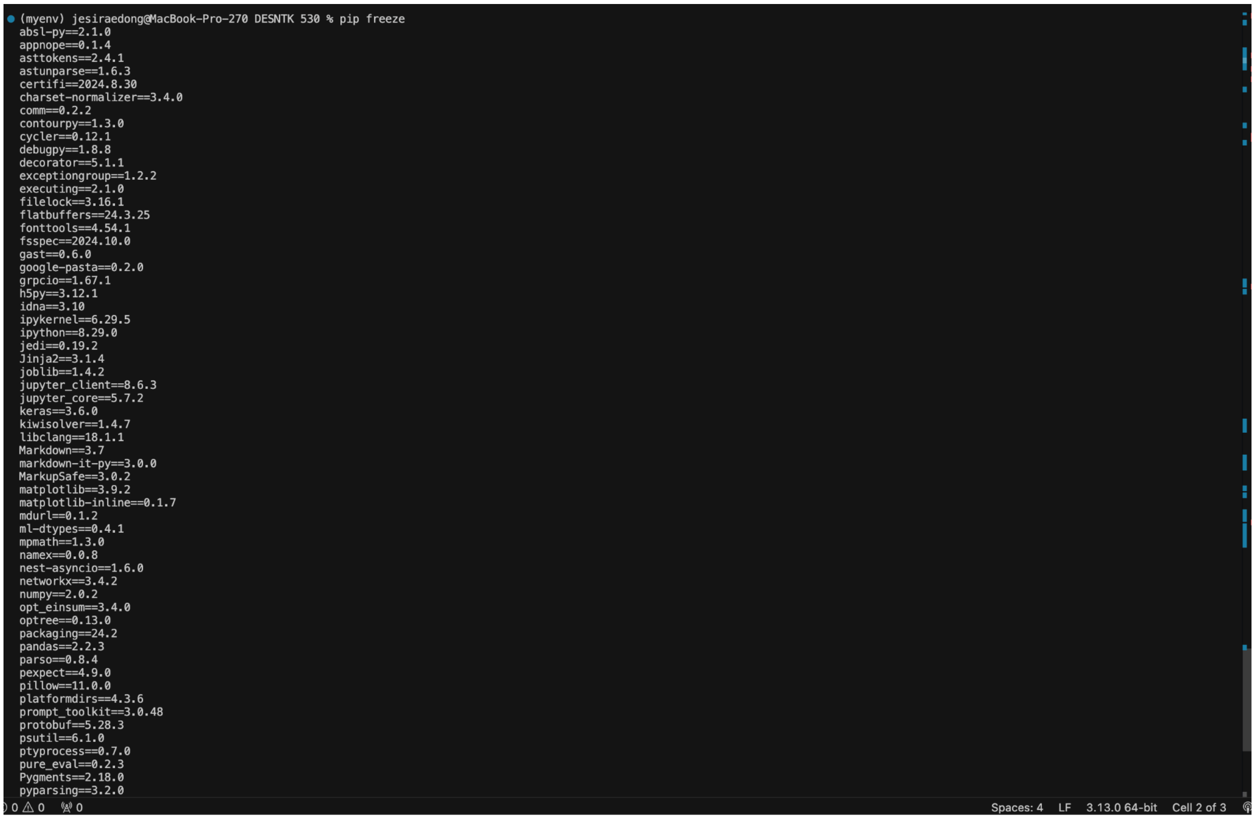
In this section, you install packages in your local development environment.

- **scikit-learn**: Essential for implementing classic ML algorithms, data preprocessing, model evaluation, and much more. Install with: `pip install scikit-learn`
- **pandas**: Vital for data manipulation and handling structured data. It's often used to load and prepare datasets.Install with: `pip install pandas`
- **numpy**: Fundamental for numerical computations, especially useful for handling arrays and matrices in ML algorithms. Install with: `pip install numpy`
- **matplotlib** and **seaborn**: Both are essential for data visualization, with `matplotlib` providing a wide range of plotting functionalities and `seaborn` offering statistical visualizations. Install with: `pip install matplotlib seaborn`
- **tensorflow** and **pytorch**: For implementing deep learning models. Choose one based on preference, as both are widely used.Install TensorFlow with: `pip install tensorflow and` Install PyTorch with: `pip install torch`

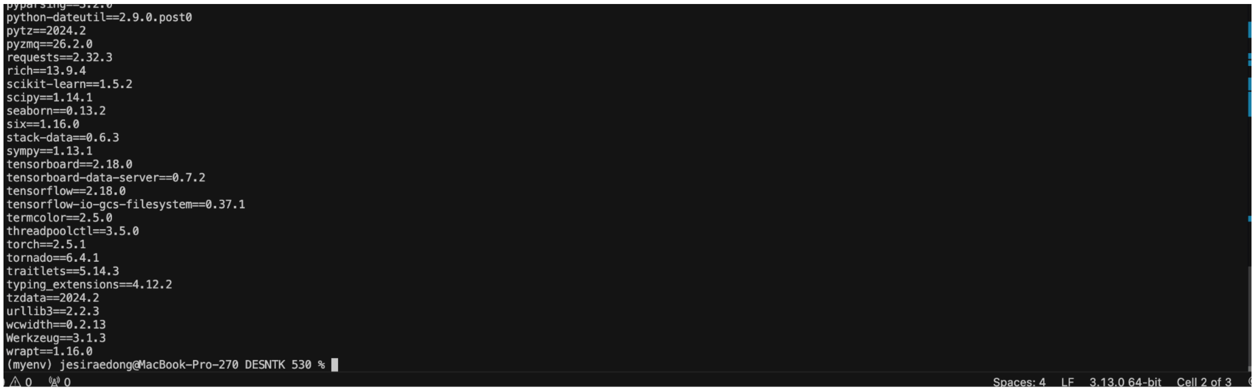
Problem 2.1: Upload screenshot of successfully installing them in your VSCode (Hint: You can use pip freeze command to know libraries in your env)



```
(myenv) jesiraedong@MacBook-Pro-270 DESNTK 530 % pip freeze
absl-py==2.1.0
astunparse==1.6.3
certifi==2024.8.30
charset-normalizer==3.4.0
contourpy==1.3.0
cycler==0.12.1
filelock==3.16.1
flatbuffers==24.3.25
fonttools==4.54.1
fsspec==2024.10.0
gast==0.6.0
google-pasta==0.2.0
grpcio==1.67.1
h5py==3.12.1
idna==3.10
Jinja2==3.1.4
joblib==1.4.2
keras==3.6.0
kiwisolver==1.4.7
libclang==18.1.1
Markdown==3.7
markdown-it-py==3.0.0
MarkupSafe==3.0.2
matplotlib==3.9.2
mdurl==0.1.2
ml-dtypes==0.4.1
mpmath==1.3.0
namex==0.0.8
networkx==3.4.2
numpy==2.0.2
opt_einsum==3.4.0
optree==0.13.0
packaging==24.2
pandas==2.2.3
pillow==11.0.0
protobuf==5.28.3
Pygments==2.18.0
pyparsing==3.2.0
python-dateutil==2.9.0.post0
pytz==2024.2
requests==2.32.3
rich==13.9.4
scikit-learn==1.5.2
scipy==1.14.1
seaborn==0.13.2
six==1.16.0
sympy==1.13.1
tensorboard==2.18.0
tensorboard-data-server==0.7.2
tensorflow==2.18.0
tensorflow-io-gcs-filesystem==0.37.1
termcolor==2.5.0
threadpoolctl==3.5.0
torch==2.5.1
typing_extensions==4.12.2
tzdata==2024.2
urllib3==2.2.3
Werkzeug==3.1.3
wrapt==1.16.0
```



```
● (myenv) jesiraedong@MacBook-Pro-270 DESNTK 530 % pip freeze
  absl-py==2.1.0
  appnope==0.1.4
  asttokens==2.4.1
  astunparse==1.6.3
  certifi==2024.8.30
  charset-normalizer==3.4.0
  comm==0.2.2
  contourpy==1.3.0
  cycler==0.12.1
  debugpy==1.8.8
  decorator==5.1.1
  exceptiongroup==1.2.2
  executing==2.1.0
  filelock==3.16.1
  flatbuffers==24.3.25
  fonttools==4.54.1
  fsspec==2024.10.0
  gast==0.6.0
  google-pasta==0.2.0
  grpcio==1.67.1
  h5py==3.12.1
  idna==3.10
  ipykernel==6.29.5
  ipython==8.29.0
  jedi==0.19.2
  Jinja2==3.1.4
  joblib==1.4.2
  jupyter_client==8.6.3
  jupyter_core==5.7.2
  keras==3.6.0
  kiwisolver==1.4.7
  libclang==18.1.1
  Markdown==3.7
  markdown-it-py==3.0.0
  MarkupSafe==3.0.2
  matplotlib==3.9.2
  matplotlib-inline==0.1.7
  mdurl==0.1.2
  ml-dtypes==0.4.1
  mpmath==1.3.0
  namex==0.0.8
  nest-asyncio==1.6.0
  networkx==3.4.2
  numpy==2.0.2
  opt_einsum==3.4.0
  optree==0.13.0
  packaging==24.2
  pandas==2.2.3
  parso==0.8.4
  pexpect==4.9.0
  pillow==11.0.0
  platformdirs==4.3.6
  prompt_toolkit==3.0.48
  protobuf==5.28.3
  psutil==6.1.0
  ptyprocess==0.7.0
  pure_eval==0.2.3
  Pygments==2.18.0
  pyparsing==3.2.0
```



```
pyparsing==3.2.0
python-dateutil==2.9.0.post0
pytz==2024.2
pyzmq==26.2.0
requests==2.32.3
rich==13.9.4
scikit-learn==1.5.2
scipy==1.14.1
seaborn==0.13.2
six==1.16.0
stack-data==0.6.3
sympy==1.13.1
tensorboard==2.18.0
tensorboard-data-server==0.7.2
tensorflow==2.18.0
tensorflow-io-gcs-filesystem==0.37.1
termcolor==2.5.0
threadpoolctl==3.5.0
torch==2.5.1
tornado==6.4.1
traitlets==5.14.3
typing_extensions==4.12.2
tzdata==2024.2
urllib3==2.2.3
wcwidth==0.2.13
Werkzeug==3.1.3
wrapt==1.16.0
(myenv) jesiraedong@MacBook-Pro-270 DESNTK 530 % █
```

3. Work with data Locally.

In this section, you will download a dataset and work with it locally in VSCode to perform a simple data analysis, using the tools of the previous module.

Problem 3.1: Select and download a dataset of your choice in `.csv` format. Once downloaded, upload the dataset to your local VS Code environment for analysis.

Problem 3.2: Load the dataset into VS Code and calculate summary statistics (mean, median and standard deviation) and create at least one exploratory data visualization. Since the file is now stored locally rather than in Google Drive, make adjustments to load and manipulate the data directly from your local storage.

Problem 3.3: After cleaning and processing the data in VS Code as you did in Google Colab, compare and contrast the experience of using VS Code with Google Colab for this workflow. Discuss the advantages and disadvantages of using VS Code for data analysis tasks, including but not limited to:

- Ease of use and setup for data loading and manipulation

  VS Code offers flexibility and can be customized with extensions for data loading and manipulation. It provides an integrated development environment with debugging, terminal, and Git integration. However, the initial setup can be complex, as you need to install Python, libraries (like pandas, numpy, etc.), and any additional packages manually. Managing virtual environments is also necessary for package dependencies. On the other hand, Colab is more beginner-friendly with zero setup for Python environments and commonly used libraries pre-installed.

- Resource management and computing power

  VS runs locally, giving us control over resources. But it requires a good local machine with sufficient CPU/GPU and RAM for intensive tasks. Limited by your hardware unless you configure remote servers. Colab provides free access to cloud-based resources, including GPUs and TPUs, which is beneficial for training deep learning models. However, there exists session timeouts and usage restrictions.

- Extension capabilities and customization options in VS Code vs. Colab's built-in features

  VS is highly customizable with a wide range of extensions, from Jupyter support to data visualization. It can be tailored for many workflows (Python, R, etc.). However, customization requires configuration and can be hard for beginners. Some complex setups, like remote server connections, may need additional configuration. Colab offers limited installing packages and mounting Google Drive. It is optimized for Jupyter-like workflows with built-in support for text, code cells, and interactive widgets. But extensions and integrations are limited to what Google Colab natively supports or allows through its UI.

- Collaboration capabilities (file sharing, notebook sharing, etc.)

Collaboration in VS is possible through extensions like Live Share, allowing real-time code sharing and editing. It also integrates with GitHub and other version control systems. Colab is better with its collaboration capabilities which are similar to Google Docs, allowing real-time notebook sharing and editing.

- Accessibility and portability across different devices and environments

  VS Code is Highly portable if set up properly, with remote SSH and cloud integration. With proper configuration, you can access it on different devices and environments, including Windows, macOS, and Linux. But files are usually stored locally unless integrated with cloud storage solutions. And cloud-based Colab is accessible from any device with a browser. However, it requires an internet connection.

4. Stretch

Problem 4.1: Upload your analysis done in the previous step to github with proper README.md file, requirements.txt file and code with proper comments. Submit your github link.